

Monitoring Social Distancing by Smart Phone App in the Effect of COVID-19

Dr. Neelavathy Pari S
Department of Computer Technology
Madras Institute of Technology
Chennai, India

Balaji Vasu
Department of Computer Technology
Madras Institute of Technology
Chennai, India

Geetha A V
Department of Computer Technology
Madras Institute of Technology
Chennai, India

Jeevitha V K
Department of Information Technology
Madras Institute of Technology
Chennai, India

Abstract—Social distancing measures are necessary for many infectious diseases that spreads through droplets and micro-droplets. According to WHO, the preventive measure for COVID-19 is to follow strict social distancing. It is not easy to enforce social distance easily in a crowded region and people often not maintain sufficient distance with neighbors. Driven by the need for energy-efficient and cost-effective social distancing monitoring, this paper proposes Smart Social Distancing (SSD) mobile application-based monitoring, which can predict the social distancing between two people assisted by mobile bluetooth and mobile camera. SSD involves two major steps to predict the social distance: first the pedestrian in the video frames is identified with the aid of Deep Learning (DL) and in the second step, distance between the two pedestrian is estimated through image processing techniques. The application can also be configured to calculate the distance using Bluetooth Low Energy (BLE) by calculating its received signal strength. The application demonstrates 85% accuracy on predicting the social distancing and alert the user using beep sound or alert message.

Keywords—Computer Vision; Bluetooth Low Energy; Machine Learning; Android App Development; Social Distancing.

I. INTRODUCTION

On 31 Dec 2019, a number of pneumonia cases of unspecified aetiology in Wuhan, was notified to the WHO China Country Office. But, the influenza identified as COVID-19 quickly spread to many countries and on 11 March 2020, the WHO declared the outbreak a global pandemic [1]. Influenza outbreaks present a possible threat to humans because of their ability to rapidly adapt and impact people who are immunologically weaker. Readiness for pandemic flu is therefore a key step in response to these outbreaks [2]. In this regard, responding to the current COVID-19 pandemic and planning for possible pandemics needs critical preparation when there is no vaccine supply is to apply social distancing, quarantine measures along with rapid delivery of medical supplies such as personal protective equipment, anti-virus treatments, etc. Some of the measures used in the past as in the case of Spanish flu of 1918 or the recent, H1N1 pandemic of 2009 was to wearing masks, maintaining personal hygiene, social distancing, contact tracing and isolation of infected individuals helped in containing the spread of the influenza to a larger extent. Thus, social distancing is a crucial measure to contain the

spread of influenza outbreaks in a region. Moreover, according a recent study [3], assessed that one-time measures would not be adequate to reduce the occurrence of COVID-19, with the current critical care facilities of United States. According to the study it is indicated that these measures may be required till 2022. Especially in countries that lacks adequate public health and testing resources, “social distancing” or “physical distancing” measure helps in flattening the transmission curve, thus aiding in reduction of burden of the overwhelmed critical care units. Therefore, an one-time or single duration social distancing measure does not suffice to avoid intensive care resources from being overrun by the COVID-19 epidemic, as it retains enough of the population to be affected by the rebound in transmission, after the end of this pandemic cycle [4].

It is estimated that the lower bound of the economic benefit of social distancing measure in United States is approximately \$8 trillion, which is greater than a third of its GDP. Moreover, the non-monetized benefits of social distancing include, quality treatment for non-COVID19 patients, short lock-down periods, buying time for developing a vaccine and strengthening the health sector resources [5]. Therefore, it is necessary to encourage appropriate social distancing.

The proposed system has two methods for monitoring social distancing and alerting the smart phone users. The first method was developed using smart phone camera which will get the feed and processed with on-device deep learning algorithm deployed in the mobile to predict the social distancing. The implementation and result of this surveillance method are discussed in below sections. The second method measures the social distance between two smart phones using Bluetooth Low Energy technique. When the user scan for nearby devices, the Received Signal Strength Indicator (RSSI) value is received during the scan callback. The distance between the other device can be calculated using the received RSSI value, the formula and algorithm used for distance calculation is discussed in below sections. If the user violates social distancing, the app is designed to alert the users by a beep sound and displaying alert message. The experiment was conducted on various places such as College entrance, Office premises, Government office, Cafeteria and

so on. 85% of accuracy is obtained on detecting the real time social distance by these two methods.

The rest of the paper is organized as follows: Section 2 explains the background and literature survey of pedestrian prediction and calculate distance between two objects in the 2D image and Bluetooth techniques. And Section 3 explains the proposed system of monitoring app used to predict the social distancing and various algorithm used for the prediction. Section 4 explains implementation and results for the developed application. Finally, in Section 5, summarizes the research conclusion and applicable feature work.

II. RELATED WORKS

In this section, three types of techniques are reviewed which is used in the proposed system: (A) Pedestrian Prediction Method, (B) Distance Calculation by Image Processing Method, (C) Bluetooth Distance Estimation Method

A. Pedestrian Prediction Method

Various methods have been developed by various authors for pedestrian detection in both 2D and 3D image space. Jianqi Zhong et al. [6] developed pedestrian motion detection in 3D image since the human motion depends on 3D space. The authors developed twin deep neural network based on pose estimation. In [7], multi-task deep learning with no action recognition is used to estimate the time for pedestrian road crossing. This detection system is developed to avoid road accidents based on movement of people who crossing the roads. Quintero et al. [8] proposed Gaussian dynamic models and Activity recognition for detection pedestrian in order to avoid roadside accidents. The proposed method is developed from multiple features of pedestrian activity such as: walking, stopping, starting and standing. Rehder et al. [9] studied pedestrian prediction using Deep Neural network which can be used in traffic management and to avoid collision in autonomous vehicle. The authors trained the model using monolithic neural network via reverse reinforcement learning. Choi et al. [10] developed Deep learning based pedestrian trajectory considering the location relationship between the people. The displacement between neighboring frames for every pedestrian in the sequential video frames are calculated. The motion information is encoded using LSTM (Low Short Term Memory) and uses MLP (Multilayer perceptron) to map the location.

B. Distance Calculation by Image Processing Method

The distance between two objects in 2D image can be calculated by different image processing techniques and mathematical formulations like Euclidean and Manhattan distance formula. Johnny Singh et al. [11] developed distance calculation of a object in 3D image using Stereo camera. The distance of the object is calculated using Euclidean distance between centroid of the bounding box and the camera and they obtained the result with 2.08 average error. JongBae [12] proposed a method to detect pedestrian and distance estimation using smart phone based thermal camera which can run in low light environment. The detector was created using multi-stage cascade learning and distance is estimated by calculating the position of the pedestrian in the 2D thermal image. On their experiment they obtained 91% accuracy in

detecting pedestrian and 85% in distance estimation. Chong Sun et al. [13] in their paper discussed distance estimation between two persons using latent variables by minimizing a distance function. They developed latent metric learning method using effective metric matrix.

C. Bluetooth Distance Estimation Method

Bluetooth low Energy (BLE) is a low cost, low power portable technology which is growing exponentially. Lam and She [14] in their paper discussed on the distance estimation of the moving object using BLE. They Received Signal Strength Indicator (RSSI) from the noisy environment and overcome by using Kalman Filter (KF) to smooth the RSSI measurement. The filtered RSSI value is trained by Support vector machine using non-linear kernel function to estimate the distance. The mobile users received limited rate for distance estimation and there is some delay on calculating the distance. Yoon et al. [15] used BLE and ultrasound signal to calculate the distance between the device. The received RSSI value are not accurate due to various environment factors. For estimating distance, the authors introduced propagation constants estimation based on Time difference of arrival between BLE and ultrasound signals. The estimated error increases as the distance increases in real time, they obtained distance estimation with 10cm average error distance. Yuan and Chang [16] proposed Indoor positioning system based on RSSI value from the BLE beacon. The authors used weighted multi point algorithm with cellular network assistance for calculating real time distance and achieved error rate of 0.8%.

III. SOCIAL DISTANCE MONITORING

For measuring social distance two methods are proposed, one is using mobile camera which can process the video frames and calculate the distance. Next method uses smart phone Bluetooth low energy BLE (Bluetooth Low Energy) hardware and calculates distance by its signal strength. Explained step by step process of both the methods in the below section.

A. Camera Surveillance Method

The smart phone camera captures the video frames to track the social distancing. Image rendered frame by frame are passed in loop in the image detection algorithm. Image processing are done for noise removal and passed to the pedestrian trained model. The outcome image will draw a rectangular bounding box when any persons are identified in the image. The classified image with person is feed to calculate distance using Euclidean algorithm. The whole working process of the system is explained in the Fig 1.

I. Predicting Pedestrian

The pedestrian prediction training model is trained by Tensorflow environment. The following steps are required in order to train the model.

- Gathering Images and Labeling data
- TF Records Creation
- Training and Exporting model

For gathering the data set, the images are downloaded from INRIA Person data set and Penn-Fudan database which different image formats are converted to JPEG format. Maintained 80% of images to training and 20% to test testing folder. The tensor flow needs more number of images in various background to train the model with good precision

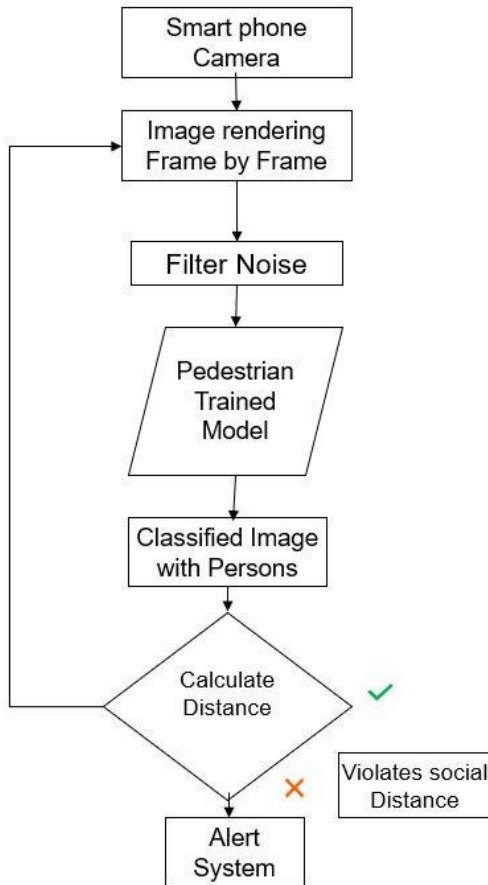


Fig. 1: System Flow

and accuracy. After downloading the data set, the labelling of all the images are done by creating a rectangle box on the target object using the tool called labellmg.

The TFRecord file is tensor flow format used for storing a sequence of binary records. After labelling process and saving all the image rectangle box as xml file, the tffile are generated for both train and test folder by exporting all the xml files into csv file. With the help of tensor flow api, the csv file is converted into tfrecord file.

The transfer learning is used for training the model which is technique of applying already trained model for training another model which can take less time to train and yeilds better results. For example knowledge gained for training

cars can be transfered to recognize trucks. The Single Shot Detection(SSD) with MobileNet pretrained model is used to train with COCO dataset called SSD model. Once the model is trained with above configuration, the model is evaluate with COCO evaluation technique by downloading COCO API. Fig 3 is the sample image used for training the pedestrian model. Single Shot Detection (SSD) is the object detection algo- rithm in real time. The Faster R-CNN uses region proposal network to create a bounding boxes and utilize the box to classify object.

Algorithm 1 Person Detection Algorithm

```

Input: Test Image from Camera feed
Output: Image with person indicators as Bounding Boxes

1: for iterator1 = [ 1: num of frames in Camera Feed] do
2:   Apply filter to image
3:   Compute integral image
4: end for
5: for iterator2 = [ 1: num of shift steps] do
6:   for iterator3 = [ 1: num of stages in classifier] do
7:     for iterator4 = [ 1: num of filters in iterator3]
8:       Filter Detection
9:       Accumulate filter outputs
10:    end for
11:    if accumulation fails per stage threshold then
12:      Reject sub-window as Person
13:    end if
14:  end for
15:  if sub window passed all three stages then
16:    Accept sub-window as Person
17:  end if
18: end for
  
```

applying convolution filters to detect objects. Conv4_3 layer is used for each cell which are capable of predicting 4 object prediction in single image frame. The Fig 2 illustrates the SSD model used with Conv5_3 along with VGG-16 layer which is used to extract feature map. After training the model, the inference graph is generated and it is exported to the mobile app as tensorflow lite(.tflite). The Algorithm 1 explains step by step logic used for pedestrian detection process, for every frame in the image it detects the persons and draw the bounding boxes.

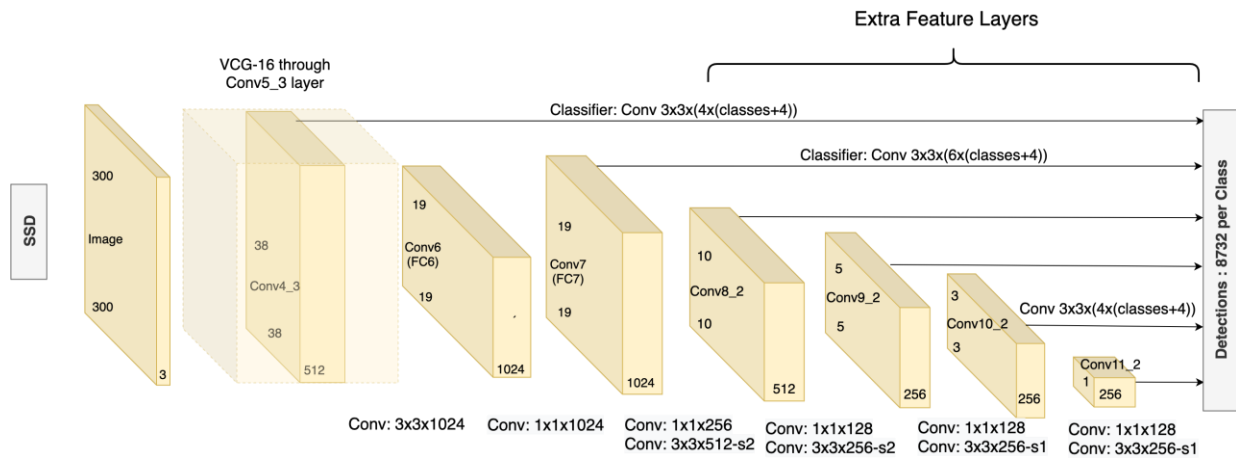


Fig. 2: Single Shot Detection Model



Fig. 3: Pedestrian Detection

II. Calculating Distance

From the frame image of pedestrian detection, the list of bounding boxes that covers every persons are retrieved. With the help of bounding boxes the center point of each is calculated using the top, left, right and bottom pixel values of bounding boxes. Then the center point of each bounding boxes are passed to distance calculation algorithm to calculate the distance between them. From the algorithm two lists of persons who maintain safe social distance and list of persons who are at unsafe distance are derived. For instance, if there is 6 persons bounding boxes in one frame as given in the Fig 4. The first person is compared with remaining all five persons distance. If the first person maintains safe distance, then the point is added in safe list. Followed with second person will be compared with other

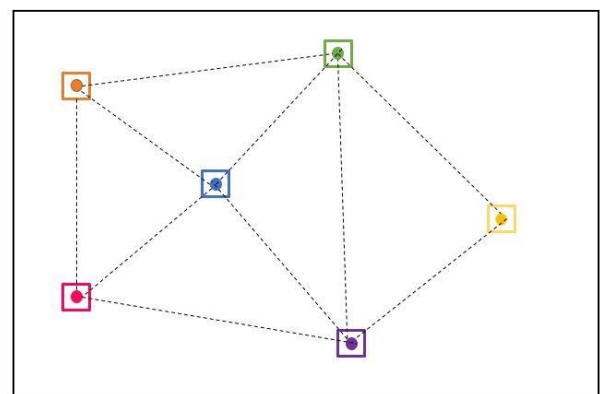


Fig. 4: Processing Bounding Boxes in Frame

Algorithm 2 Social Distance Measurement Algorithm

Input: An array of point (x, y) such that x is the x-coordinate of the point and y is the y-coordinate of the point

Output: Array of points with unsafe distance

```

1: if P.length < 2 then
2:   return nil
3: end if
4: for iterator1 = [P.length : 1] do
5:   for iterator2 = [1 : < iterator1] do
6:     D=ComputeEuclideanDistance(P[iterator1],
7:                               P[iterator2]) / AvgPersonWidth
8:     if (D < safeDistance) then
9:       addUnsafePair(P[iterator1], P[iterator2])
10:    end if
11:   end for
12: end for
13: return unsafePair
    
```

four persons excluding the first one, if the second person doesn't maintain social distance with any other points then it is added to unsafe list. Similarly, the other iterations of

persons are followed with remaining persons in the list. With the help of this technique at the end the number of safe and unsafe list data structure. The above algorithm is processed in every frames of the camera feed which is received after processing from object detection algorithm. Redraws the bounding boxes with predicted persons with safe and unsafe with different color in indication. The Algorithm 2 explains the algorithm for measuring social distancing using euclidean distance. With every frame received from camera feed with rectangular frames, the midpoint of every rectangle is calculated, and the two points are passed to measure the distance.

III. PERCENTAGE CALCULATION

Algorithm 3 Percentage and Zone Calculation

Input: An array of safe and unsafe Points
Output: Percentage and Zone of Social Distancing
1: **if** unSafeList.length < 0 **then**
2: **return** nil
3: **end if**
4: percentage = calculatePercent(safeList, unSafeList)
5: **if** percentage in safeThersold **then**
6: color = Green
7: **else if** percentage in averageThersold **then**
8: color = Orange
9: **else if** percentage in dangerThersold **then**
10: color = Red
11: **end if**

Once the distance of every bounding boxes are calculated, the list of safe and unsafe distance are obtained. To alert the pedestrians who doesn't follow social distancing, the alert window mentioning the percentage of people following safe distance and color indication of zone whether the region is in safe zone (Green Indication), area which is likely to safe and danger which means average group will fall under this category (Orange Indication) and finally unsafe zone where most of the people doesn't maintain social distance (Red Indication) are displayed in order to create alertness and awareness to the users.

Algorithm 3 explains the algorithm for percentage calculation based on the safe and unsafe distance determined. The app user is alerted by displaying various color zone whether the peoples are maintaining safe or unsafe distance.

B. Bluetooth Module

Another proposed method of calculating the distance between the two persons using smart phone is Bluetooth low energy technique. When the two persons installed this app with the help of Bluetooth low energy support 4.0 the approximate distance can be calculated. RSSI (Received Signal Strength Indicator) is the strength of the beacon signal received at smart phone. The RSSI value ranges from -26 (few inches) to -100 (40-50 meters) distance. Along with RSSI there is an- other factor called Measured power used for the calculation.

Algorithm 4 BLE Distance Calculation

Input: Received RSSI Value on scanning devices.

Output: Estimated real time distance between two smart phones

```
1: if BLE Not Supported then
2:   return nil
3: end if
4: Scan the nearby devices
5: for iterator1 = [1 : devices.length] do
6:   AdvertiseMessage()
7:   D = calculateDistance(device.rssi)
8:   realTimeDistance = findAverage(D)
9:   if realTimeDistance > thersold then
10:    displayAlertMessage()
11:  end if
12: end for
```

The formula for the distance calculation using BLE is given below.

$$Distance = 10^{((Measured\ Power - RSSI) / (10 * N))}$$

IV. IMPLEMENTATION AND RESULT

A. Experimental SetUp

For Pedestrian prediction and distance calculation experiments were conducted in Windows 10 machine (2.60 GHz, i7 processor, 32 GB RAM and 1GB Graphics Card) for training the model and exporting the model in smart phone. Following tools like TensorFlow, OpenCV and other supporting libraries are installed for training the model. Once the model is trained and exported to tensor flow lite model with the help of use Android Studio developed the android application. After developing the app it is tested in Samsung S7 Android device with Oreo supported. The results of the camera surveillance method are discussed in below sections. For the Bluetooth distance calculation, developed and installed the app in different smart phones with BLE (Bluetooth Low Energy) supported. When two smart phones are bring to closer distance alert message and sound will be popped in both the phones.

TABLE 1: Performance Results

Accuracy	Precision	Recall	F1-Score
91.10	92.5	83.06	85.3

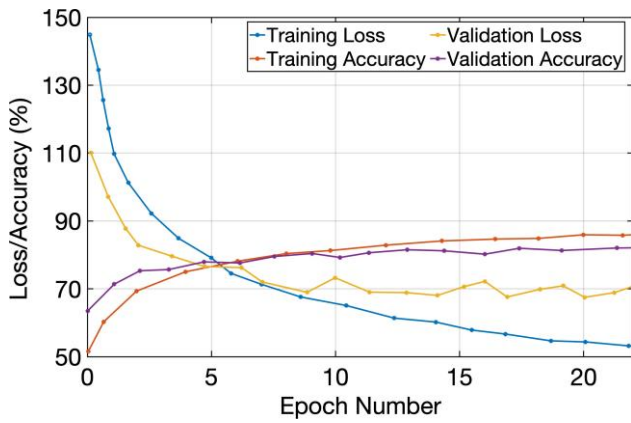


Fig. 5: Training and Validation Evaluation

B. Training and Evaluating Model

The proposed system is trained and evaluated on INRIA Person data set and Penn-Fudan datasets which contains more real-world human image data sets. This data set provides pedestrian bounding boxes for pedestrian detection and pedestrian attributes for estimating the pedestrian behavior. In order to test the detection performance, various experiments are conducted in different places. In the first detection experiment, the performances of the algorithm are investigated using the classical approach without action recognition on the INRIA data set. The Table 1 shows the accuracy, Precision and F1-Score obtained during training the model. The Fig 5 shows the accuracy and loss on training and validation of the model.

C. Pedestrian Prediction Result

The TensorFlow lite model is exported for Android mobile app development, with the help of TensorFlow SDK and API the object detection are predicted in live camera feed. Once the persons are identified with their bounding boxes it is passed to distance calculation algorithm module. The Fig 6 displays the persons predicted from the trained model along with the distance calculation determining whether they are safe or risk position. The experiment was conducted with group of people with using the model, it detects the pedestrian with 91% of accuracy. The limitation of using mobile phone camera is that it is capable of only covering small distance.

D. Distance Calculation Result

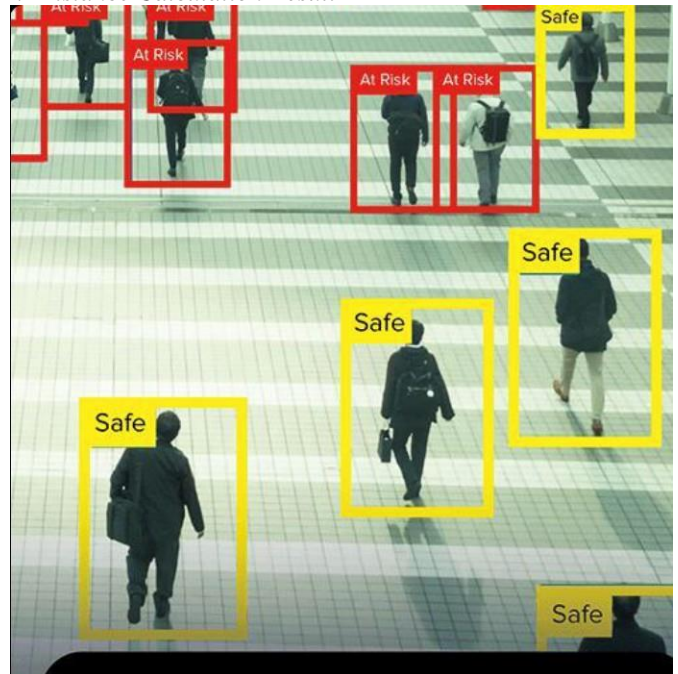


Fig. 6: Pedestrian Result

For distance calculation in image, the center point of every bounding boxes from the object detection result are obtained and distance between the two mid points are calculated. The distance are in pixels value, for real time determination of distance value, the pixels value are processed with some image processing technique. The whole pixel distance is divided with average human width in order to get exact distance in real values considering different zoom levels of camera. Once the distance are calculated a line is drawn between the two persons for indication, the green color indicated the two persons maintains the safe distance and red color indicates the two persons are at risk. The Fig 7 shows the mobile app with persons who maintains safe and unsafe distance. This experiment was conducted in public places, where people usually roams at high density and identified the social distance prediction is about 85% accurate as real values.

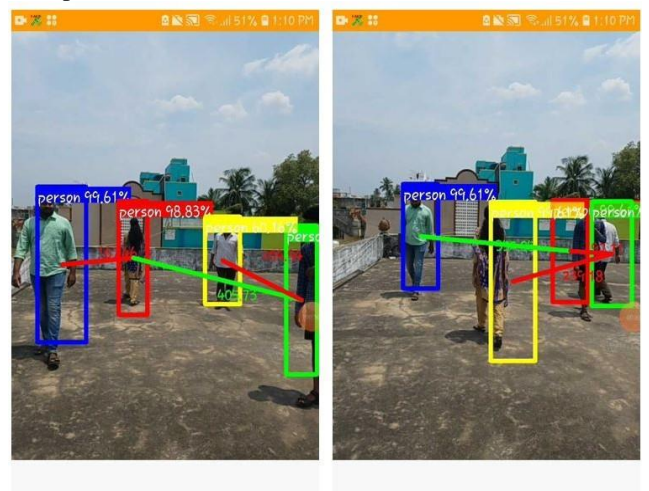


Fig. 7: Distance Calculation Result

E. Bluetooth Module Result

For distance calculation using Bluetooth low energy, developed another mobile app feature which has option to turn on social distance monitoring. When user turns on, the scanning for BLE will be triggered where the device discovery for other bluetooth device will take place. Once the new device is identified the device details like UUID, RSSI value are identified in the device discovery callback. With the help of RSSI signal strength, the distance are calculated using the formula. The App is installed in two devices, when they are brought to closer distance with less than one meter, alerts are displayed in both the phones. The Fig 8 shows the popup between two android devices which are less than one meter. The implemented mobile application estimates the distance calculation of the proposed system on two android phones keeping with measured distance say 1, 2 and 3 meters. The estimated value has the error value between 10 to 20 cm on average comparing to its real time value. The error value increases because the RSSI value have more fluctuation when the devices are at more distance due to environmental distractions. The Table 2 shows the results obtained from various phone which returns the approximate value of real time distance. The different values of rssi value are collected and determine its average value for final distance calculation. The values vary some device to device based on the hardware of Bluetooth signal receiver. The Fig 9 shows the comparative graph of actual distance and estimated distance in various mobile distance.

TABLE 2 : Distance Calculation on RSSI value

S No	Mobile Model	1M	2M	3M
1	Samsung S7	1.18	1.98	2.84
2	One Plus 5	1.04	2.10	3.02
3	Honor Lite 9	1.5	2.19	3.21
4	Moto E Plus	0.98	1.86	2.78
5	Samsung S10	1.06	2.05	3.10



Fig. 8: BLE Distance violation alert in two Android devices

V. CONCLUSION AND FUTURE WORK

The paper presents two methods of monitoring social distancing for smart phone users. In camera surveillance method, pedestrian detection is implemented using deep

learning algorithm and distance between two persons using euclidean formula. In Bluetooth distance calculation, calculated the real time distance by receiving its signal strength between two android devices. It is cost effective approach since this system is implemented in mobile app people don't

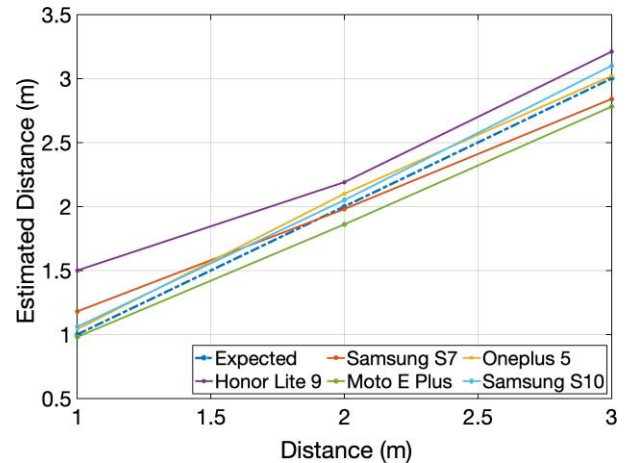


Fig. 9: Distance Measurement for Various Mobiles

want to spend cost for other infrastructure for surveillance and monitoring. Developed this model in Android mobile, as a future work it will be extended in iOS environment as well for supporting iPhone users. As discussed, the smart phone camera are capable of covering only small distance, the first approach can be leveraged in all public surveillance camera for social distance monitoring.

ACKNOWLEDGMENTS

The author would like to thank Tourenathan-G5organisation for downloading the transfer learning model from their Github link. The authors would also like to extend our thanks to Madras Institute of Technology campus, Chennai and their staffs for their help and support in making this work possible during this pandemic situation.

REFERENCES

- [1] J. A. Lewnard and N. C. Lo, "Scientific and ethical basis for social-distancing interventions against covid-19," *The Lancet. Infectious diseases*, 2020.
- [2] C. Franco-Paredes, P. Carrasco, and J. I. S. Preciado, "The first influenza pandemic in the new millennium: lessons learned hitherto for current control efforts and overall pandemic preparedness," *Journal of immune based therapies and vaccines*, vol. 7, no. 1, p. 2, 2009.
- [3] S. M. Kissler, C. Tedijanto, M. Lipsitch, and Y. Grad, "Social distancing strategies for curbing the covid-19 epidemic," *medRxiv*, 2020.
- [4] M. Kretzschmar, G. Rozhnova, and M. van Boven, "Isolation and contact tracing can tip the scale to containment of covid-19 in populations with social distancing," *Available at SSRN 3562458*, 2020.
- [5] M. Greenstone and V. Nigam, "Does social distancing matter?," *University of Chicago, Becker Friedman Institute for Economics Working Paper*, no. 2020-26, 2020.
- [6] J. Zhong, H. Sun, W. Cao, and Z. He, "Pedestrian motion trajectory prediction with stereo-based 3d deep pose estimation and trajectory learning," *IEEE access*, vol. 8, pp. 23480–23486, 2020.
- [7] D. O. Pop, A. Rogozan, C. Chatelain, F. Nashashibi, and A. Ben-shair, "Multi-task deep learning for pedestrian detection, action recognition and time to cross prediction," *IEEE Access*, vol. 7, pp. 149318–149327, 2019.

- [8] R. Q. Mínguez, I. P. Alonso, D. Fernández-Llorca, and M. Á. Sotelo, "Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1803–1814, 2018.
- [9] E. Rehder, F. Wirth, M. Lauer, and C. Stiller, "Pedestrian prediction by planning using deep neural networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–5, IEEE, 2018.
- [10] I. Choi, H. Song, and J. Yoo, "Deep learning based pedestrian trajectory prediction considering location relationship between pedestrians," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIC)*, pp. 449–451, IEEE, 2019.
- [11] N. J. Singh and K. Nongmeikapam, "Stereo system based distance calculation of an object in image," in *2019 Fifth International Conference on Image Information Processing (ICIIP)*, pp. 29–34, IEEE, 2019.
- [12] J. Kim, "Pedestrian detection and distance estimation using thermal camera in night time," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIC)*, pp. 463–466, IEEE, 2019.
- [13] C. Sun, D. Wang, and H. Lu, "Person re-identification via distance metric learning with latent variables," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 23–34, 2016.
- [14] C. H. Lam and J. She, "Distance estimation on moving object using ble beacon," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–6, IEEE, 2019.
- [15] S.-U. Yoon, J. Woo, J. Cho, and C.-R. Rahm, "An efficient distance estimation method based on bluetooth low energy and ultrasound," in *2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 206–212, IEEE, 2018.
- [16] Y. You and C. Wu, "Indoor positioning system with cellular network assistance based on received signal strength indication of beacon," *IEEE Access*, 2019.
- [17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.
- [18] T. Rosman, A. Chasiotis, M. Kerwer, H. Steinmetz, O. Wedderhoff, and M. Bosnjak, "Will covid-19-related economic worries superimpose the health worries, reducing acceptance of social distancing measures? a prospective pre-registered study," 2020.
- [19] X. Zhu, X.-Y. Jing, X. You, X. Zhang, and T. Zhang, "Video-based person re-identification by simultaneously learning intra-video and inter-video distance metrics," *IEEE Transactions on Image Processing*, vol. 27, no. 11, pp. 5683–5695, 2018.
- [20] B. Galdino and A. Nicolau, "A measure distance system for docks: An image-processing approach," in *2017 IEEE First Summer School on Smart Cities (S3C)*, pp. 145–148, IEEE, 2017.
- [21] E. DANDIL and K. K. ÇEVİK, "Computer vision-based distance measurement system using stereo camera view," in *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1–4, IEEE, 2019.
- [22] M. Ji, J. Kim, J. Jeon, and Y. Cho, "Analysis of positioning accuracy corresponding to the number of ble beacons in indoor positioning system," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*, pp. 92–95, IEEE, 2015.
- [23] N. Toure, *Training your Object Detection model on TensorFlow (Part 2)*.
- [24] *Penn-Fudan Database for Pedestrian Detection and Segmentation*.
- [25] N. Toure, *Convert a TensorFlow frozen graph to a TensorFlow lite (tflite) file (Part 3)*.
- [26] MChhab, *Android – Calculating Distance Between Two Points*, 2009 (accessed February 3, 2014).
- [27] M. Senn, *Using L^AT_EX for Your Thesis*, 2009 (accessed February 3, 2014).
- [28] S.-H. Tsang, *Review: SSD — Single Shot Detector (Object Detection)*.